Lecture no. 2

Shibboleth SP: installation and basic configuration for Single Sign On (SSO) second part
Outline

- Description of user session: attributes
- How to read session data
  - PHP example
  - Python example
  - Java example using Apache+Tomcat
- Passive session (lazy session)
Anatomy of a federated login

- User requests a federated resource (SP)
- User is redirected to the login page of the IdP
- IdP exchanges the authenticated user attributes with the SP (Name, Surname, email, etc.)
- IdP redirects the authenticated user to the resource
Anatomy of a User session

- A Shibboleth user session in Apache HTTPD consists of the following elements:
  - `REMOTE_USER`, a special Apache variable which contains the username of the authenticated user
  - Some user attributes, inserted in the Apache environment
Which attributes?

- User attributes are technically described in an LDAP schema. The most common are:
  - LDAPv3
  - rfc4519
  - inetorgPerson
  - Schac
  - eduPerson

- Most federations, e.g. IDEM, release guidelines on which attributes can be used in IdPs and SPs. This may vary from federation to federation. They may include
  - **Personal characteristics** (Name, surname, title)
  - **Contact details** (email, telephone, organisation)
  - **Authorisation and accounting** (affiliation, entitlement)
Example: Name and Surname

- Some **information** may be contained in **different attributes**
- Take for example name and surname, they are contained in
  - displayName
  - commonName
  - givenName and sn
- When you have to use these attributes, it's the “**search**” for the **information in the various attributes** before good practice to which may contain them.
REMOTE_USER Configuration

- The specific attributes of the Shibboleth session which need to be used for the REMOTE_USER are specified on the SP in the file
  /etc/shibboleth/shibboleth2.xml

- The first attribute that the Shibboleth session finds is used
  
  ```xml
  <ApplicationDefaults
  entityId="https://sp1.local/shibboleth"
  REMOTE_USER="eppn persistent-id targeted-id">
  
  The REMOTE_USER is therefore nothing more than a special Shibboleth session attribute
On the SP, in the file
/etc/shibboleth/attribute-map.xml
uncomment the part
<!-- Some more eduPerson attributes, uncomment these to use them... -->
<Attribute name="urn:mace:dir:attribute-def:eduPersonPrimaryAffiliation"
id="primary-affiliation">
<AttributeDecoder xsi:type="StringAttributeDecoder"
caseSensitive="false"/>
</Attribute>
<Attribute ...>
<Attribute ...
<Attribute ...
<Attribute ...
Reading attributes: PHP

- Apache environment variables are read by PHP from the $_SERVER dictionary

```php
<?php
function getName() {
    if (array_key_exists("displayName", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["displayName"]));
    } else if (array_key_exists("cn", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["cn"]));
    } else if (array_key_exists("givenName", $_SERVER) && array_key_exists("sn", $_SERVER)) {
        return implode(" ", explode(";", $_SERVER["givenName"])) . " ".
            implode(" ", explode(";", $_SERVER["sn"]));
    }
    return "Unknown";
}
$username = $_SERVER["REMOTE_USER"];
$name = getName();
print "<h1>Hi " . $username . "!!!</h1>";
print "<p>Your name is; " . $name . ".</p>";
?>
```
Environment variables are read in Python via the `os.environ` dictionary

```python
#!/usr/bin/python2.7
import cgitb
from os import environ
cgitb.enable()
def get_name():
    if "displayName" in environ:
        return " ".join(environ["givenName"].split(";"))
    elif "cn" in environ:
        return " ".join(environ["cn"].split(";"))
    elif "givenName" in environ and "sn" in environ:
        return " ".join(environ["givenName"].split(";")) + " " + " " ".join(environ["sn"].split(";"))
    return "Unknown"

print "Content-Type: text/html;charset=utf-8\n"
username = environ.get("REMOTE_USER", None)
name = get_name()
print "<h1>Hi %s!!!</h1>" % username
print "<p>Your name is; %s.</p>" % name
```
Reading attributes : Java/schema

- The preferred configuration for Java applications is to use Apache httpd as a frontend to Tomcat and have them communicate using an **AJP connector**
The AJP connector automatically passes the Apache environment variables to the Tomcat server, but ONLY those with prefix AJP_

In the file
/etc/shibboleth/shibboleth.xml

<ApplicationDefaults
entityID="https://sp1.local/shibboleth"
REMOTE_USER="eppn persistent-id targeted-id"
attributePrefix="AJP_">
The AJP connector automatically passes the REMOTE_USER but, for Tomcat to read it, it is necessary to disable Tomcat authentication:

In /etc/tomcat7/server.xml:

```xml
<Connector port="8009" protocol="AJP/1.3" redirectPort="8443"
  tomcatAuthentication="false" />
```
Now we can read the variables in a servlet from the request object as follows:

```java
private String getName(HttpServletRequest request) {
    if (request.getAttribute("givenName") != null) {
        return (String) request.getAttribute("givenName");
    } else if (request.getAttribute("cn") != null) {
        return (String) request.getAttribute("cn");
    } else if (request.getAttribute("givenName") != null && request.getAttribute("sn") != null) {
        return (String) request.getAttribute("givenName") + " " + (String) request.getAttribute("sn");
    }
    return "Unknown";
}
```

protected void doGet(HttpServletRequest request, HttpServletResponse response) throws ServletException, IOException {
    PrintWriter pw = response.getWriter();
    String username = request.getRemoteUser();
    String name = getName(request);
    pw.println("<h1>Hi " + username + "!!!</h1>");
    pw.println("<p>Your name is " + name + "</p>");
```
The “lazy session”

- For some applications, it may make sense to offer **some functionality to anonymous users**, and allow **authenticated users** to perform **additional functions**.

- In other cases, applications could be accessible **both via federated access and local credentials** (federated or otherwise).

- In these cases, Shibboleth offers the so-called **“lazy session”**.
Example of lazy session
The configuration of the Location in Apache has to be changed as follows:

- In `/etc/apache2/sites-enabled/service_provider.conf`

```xml
<Location /lazy.php>
  AuthType shibboleth
  ShibRequestSetting requireSession false
  Require shibboleth
</Location>
```
Lazy session: example page

- We can now create a PHP page with lazy authentication (e.g. /lazy.php)

```php
<?php
$l = "/Shibboleth.sso/Login?target=/lazy.php";
$username = $_SERVER["REMOTE_USER"];

if (!isset($username) || empty($username)) {
    print "<p>Anonymous user ";
    print "<a href="".$l."">Login</a></p>";
} else {
    print "<p>Authenticated user: ".$username."</p>;
}
?>
```
Any questions?
Configuring Session Attributes

- Session Attributes have to be configured on both IdP side and SP side

- In both cases, there are two phases:
  - **Mapping and retrieving** the values of the session attributes (attribute-resolver, attribute-map)
  - **Filtering and removal** of unwanted or non-conforming attributes (attribute-filter, attribute-policy)
Example of IdP configuration

- In `/opt/shibboleth-idp/conf/attribute--resolver.xml`

```xml
<resolver:AttributeDefinition id="givenName"
   xsi:type="ad:Simple" sourceAttributeID="givenName">
   <resolver:Dependency ref="myLDAP" />
   ...
   <resolver:AttributeEncoder xsi:type="enc:SAML1String"
       name="urn:mace:dir:attribute-def:givenName" />
   <resolver:AttributeEncoder xsi:type="enc:SAML2String"
       name="urn:oid:2.5.4.42" friendlyName="givenName" />
</resolver:AttributeDefinition>
```

- In `/opt/shibboleth--idp/conf/attribute--filter.xml`

```xml
<afp:AttributeRule attributeID="givenName">
   <afp:PermitValueRule xsi:type="basic:ANY"/>
</afp:AttributeRule>
```
Example of SP configuration

In /etc/shibboleth/attribute-map.xml

```
<Attribute name="urn:mace:dir:attribute-def:givenName" id="givenName"/>
<Attribute name="urn:oid:2.5.4.42" id="givenName"/>
```

In /etc/shibboleth/attribute-policy.xml

```
<afp:AttributeRule attributeID="givenName">
    <afp:PermitValueRule xsi:type="ANY"/>
</afp:AttributeRule>
```
Any questions?